VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY
INSTITUTE OF MATHEMATICS

STATISTICKÉ ZPRACOVÁNÍ ROZSÁHLÝCH DAT VE SVOZOVÝCH ÚLOHÁCH
STATISTICAL EVALUATION OF LARGE-SCALE DATA OF WASTE COLECTION PROBLEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Zlata Šmídová

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Radovan Šomplák, Ph.D.

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav matematiky
Studentka:	**Zlata Šmídová**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	**Ing. Radovan Šomplák, Ph.D.**
Akademický rok:	2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Statistické zpracování rozsáhlých dat ve svozových úlohách

Stručná charakteristika problematiky úkolu:

Prace bude zaměřena na zpracování rozsáhlých souborů dat z reálného provozu svozových společností v oblastech odpadového hospodářství.
Úkolem studenta bude data zpracovat a statisticky vyhodnotit pro účely následné optimalizace provozu řešené na Ústavu procesního inženýrství. V rámci práce si student prohloubí znalosti matematické statistiky a pro práci s daty využije vhodný statistický software (např. STATISTICA). V průběhu zpracování bude student konzultovat práci se specialisty na Ústavu matematiky, odbor statistiky (Dr. Josef Bednář). Úloha je motivována činnostmi strategického projektu TAČR "Centrum kompetence pro energetické využití odpadů" řešeného na Ústavu procesního inženýrství a přípravou projektu SMART CITY.

Cíle bakalářské práce:

- Student navrhne vhodnou metodu zpracování velkého množství dat z reálného provozu s využitím metod popisné a výběrové statistiky.
- Student si prohloubí poznatky z oblasti statistického zpracování rozsáhlých dat, viz data mining.
- Dále bude vybrán způsob archivace dat s možností opětovného statistického zpracování.
- Hlavním výstupem práce bude popsání základních vlastností dat, které budou sloužit jako vstupní data do úloh typu "VRP".

Seznam doporučené literatury:

ANDĚL, Jiří.1998. Statistické metody. Praha: Matfyzpress, 2. vyd.

ZVÁRA, Karel a Josef ŠTĚPÁN. 2002. Pravděpodobnost a matematická statistika. Praha: Matfyzpress, 2. vyd.

SON, Lehoang and Amal LOUATI. 2016. Modeling municipal solid waste collection: A generalized vehicle routing model with multiple transfer stations, gather sites and inhomogeneous vehicles in time windows. Waste Management Volume 52, pp. 34–49.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc. ředitel ústavu	doc. Ing. Jaroslav Katolický, Ph.D. děkan fakulty

Abstrakt
Bakalářská práce se zabývá statistickým zpracováním rozsáhlých dat v dopravních úlohách a vyhodnocením těchto dat pro účely následné optimalizace provozu. Statistické testy byly prováděny v programech Microsoft Excel a STATISTICA. Po sestavení matematického modelu byla zpracovaná data nahrána do databáze SQLite a do programu GAMS (General Algebraic Modeling System), který vypočítal dobu strávenou na jednotlivých úsecích tras. Výsledky jsou důležité pro řešení dopravních a logistických problémů, kterými se zabývá mnoho firem a společností. Uvedený přístup představuje novou techniku pro tvorbu okrajových podmínek v dopravních úlohách. Výstupem jsou kvalitní vstupní data pro optimalizaci v logistice.

Summary
The bachelor thesis deals with statistical evaluation of large-scale data of waste collection problem and with evaluation these data for the purpose of subsequent traffic optimization. Statistical tests were performed in Microsoft Excel and STATISTICA. After compiling the mathematical model, the processed data were uploaded to the SQLite database and to the General Algebraic Modeling System, which calculated the time spent on each section of the route. The results are important for dealing with traffic and logistics issues that many companies and companies are engaged in. This approach represents a new technique for creating boundary conditions in traffic tasks. The output is high quality input data for optimization in logistics.

Klíčová slova
dopravní úlohy, teorie grafů, matematický model, optimalizace, statistické testy, Grubbsův test, Dean-Dixonův test

Keywords
traffic tasks, graph theory, mathematical model, optimization, statistical tests, Grubbs' test, Dean-Dixon's test

ŠMÍDOVÁ, Z.*Statistické zpracování rozsáhlých dat ve svozových úlohách.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 36 s. Vedoucí Ing. Radovan Šomplák, Ph.D.

Prohlašuji, že svou bakalářskou práci na téma Statistické zpracování rozsáhlých dat ve svozových úlohách jsem zpracovala samostatně pod vedením vedoucího bakalářské práce Ing. Radovana Šompláka, Ph.D. a že jsem uvedla všechny použité materiály, ze kterých jsem čerpala.

Zlata Šmídová

Ráda bych poděkovala svému vedoucímu Ing. Radovanu Šomplákovi, PhD. za vedení, cenné rady a připomínky při zpracování této bakalářské práce. Dále bych chtěla poděkovat Ing. Vlastimíru Nevrlému za poskytnutý čas a pomoc při tvorbě práce a Ing. Tomáši Lipovskému za pomoc při práci s databázemi. Další poděkování patří Ing. Josefu Bednářovi, Ph.D. za konzultaci statistické části práce. V neposlední řadě děkuji své rodině za podporu během celého studia.

Zlata Šmídová

Obsah

1. Úvod

Tato bakalářská práce se zaměřuje na zpracování rozsáhlých souborů dat z reálného provozu dopravních společností. Na Ústavu procesního inženýrství (ÚPI) se řeší problémy optimálního nakládání s odpadem, tj. celkového zpracovatelského řetězce (sběr, přeprava a zpracování odpadu), viz [15]. Finanční náklady jsou spojené nejen se zpracováním odpadu, ale také s jeho převozem. Z tohoto důvodu je důležité data související s dopravou analyzovat (statisticky vyhodnotit) a vytvořit vstupy pro následnou optimalizaci spojenou s dopravou.

Výsledky práce nemusí být vhodné pouze pro odpadové hospodářství. Dopravní a logistické problémy řeší mnoho firem, proto by se statistické zpracování dat mohlo hodit také pro poštovní služby [41], plány tras školních autobusů [44], plány tras městské hromadné dopravy [26], úpravu silnic [30] a mnoho dalších.

Vstupní data pro analyzovaný problém pocházejí z GPS (Global Positioning System) zařízení různých dopravních společností. Tyto informace se liší v detailu zaznamenávání, tj. v prodlevě mezi jednotlivými záznamy v průběhu přepravy. Cílem práce bylo odhadnout čas průjezdu jednotlivých úseků tras, což je přínosné pro již zmíněnou optimalizaci spojenou s dopravou. K vyřešení tohoto problému bylo nutné sestavit matematický model v prostředí modelovacího systému GAMS. Matematický model je součástí Aplikační části práce, kam byl zařazen i ilustrativní příklad a případová studie.

V následujícím textu se nejprve seznámíme se základními pojmy z oblasti teorie grafů, následně bude zmíněna optimalizace. Dva různé druhy testů normality a dva na odlehlá pozorování budou popsány v kapitole, která se věnuje statistice. V této kapitole najdeme také zmínku o shlukové analýze, jež bude využita pro statistické zpracování dat.

V další části textu jsou popsány programy a nástroje, které byly dále využívány. Stěžejní v této části práce je popis propojení databáze SQLite a systému GAMS, které bylo potřeba pro nahrání všech potřebných dat.

2. Teorie grafů

V této kapitole se budeme zabývat oblastí diskrétní matematiky, a to teorií grafů. Základ diskrétní matematiky (konkrétněji teorie grafů) položil v 18. století Leonard Euler. Moderní podoba však vznikla až s nástupem počítačů a rozvojem informatiky v druhé polovině 20. století. [20]
Nyní si definujeme základní pojmy. Čerpat budeme z [9], [16], [21] a [33].

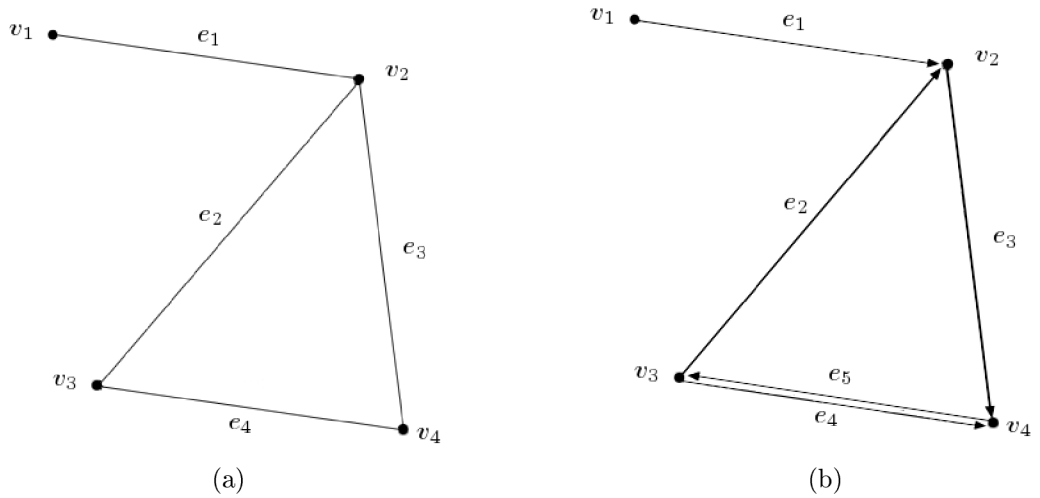
2.1. Základní pojmy

Definice 2.1 *Graf G* je uspořádaná dvojice $G = (V, E)$, kde V je neprázdná množina *vrcholů* (anglicky vertex) a E je množina dvoubodových podmnožin množiny V, tzv. množina *hran* (edge).

Definice 2.2 *Neorientovaný graf* je trojice $G = (V, E, \epsilon)$, která je tvořená neprázdnou konečnou množinou *vrcholů* V, konečnou množinou *neorientovaných hran* E a zobrazením $\epsilon : E \to V \times V$ označovaným jako *vztah incidence*. Toto zobrazení přiřazuje každé hraně $e \in E$ jednoprvkovou nebo dvouprvkovou množinu vrcholů. Je-li $\epsilon(e)$ jednoprvková množina, nazýváme hranu e *neorientovanou smyčkou*. Neorientovaný graf, který nemá smyčky, nazýváme *neorientovaným grafem bez smyček*.

Neorientovaný graf je zobrazen na obr. 2.1 (a).

Definice 2.3 *Orientovaný graf* (viz obr. 2.1 (b)) je trojice $G = (V, E, \epsilon)$, která je tvořená neprázdnou konečnou množinou *vrcholů* V, konečnou množinou *orientovaných hran* E a zobrazením $\epsilon : E \to V \times V$ označovaným jako *vztah incidence*. Toto zobrazení přiřazuje každé hraně $e \in E$ uspořádanou dvojici vrcholů *(x, y)*. Vrchol x nazveme počátečním vrcholem hrany e a značíme jej *PV(e)*, vrchol y je koncovým vrcholem hrany e a značíme jej *KV(e)*. Jestliže *PV(e)=KV(e)*, pak hranu e nazýváme *orientovanou smyčkou*.



(a) (b)

Obrázek 2.1: (a) neorientovaný graf, (b) orientovaný graf

Definice 2.4 Nechť G je orientovaný graf. Posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, v_2, ..., e_k, v_k$ nazýváme *orientovaným sledem*, jestliže pro každou hranu e_i z této posloupnosti platí $PV(e_i) = v_{i-1}$ a $KV(e_i) = v_i$.

Posloupnost vrcholů a hran $v_0, e_1, v_1, e_2, v_2, ..., e_k, v_k$ nazýváme *neorientovaným sledem*, jestliže každá hrana e_i z této posloupnosti spojuje vrcholy v_{i-1}, v_i.

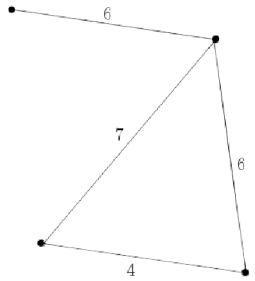
Z definice 2.4 vyplývá, že každý orientovaný sled je zároveň i neorientovaným sledem.

Definice 2.5 Orientovaný (neorientovaný) sled, v němž se žádná hrana nevyskytuje víckrát, se nazývá *orientovaný (neorientovaný) tah*.

Orientovaný (neorientovaný) sled, v němž se žádný vrchol nevyskytuje víckrát, se nazývá *orientovaná (neorientovaná) cesta*.

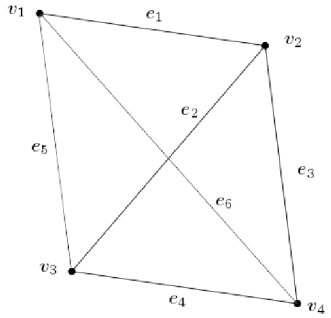
Samotné grafy často nestačí k popisu určité situace nebo systému. Využíváme proto tzv. ohodnocené grafy.

Definice 2.6 Graf, kde jsou hrany, resp. vrcholy, opatřeny číselnými hodnotami, nazýváme *ohodnoceným grafem* nebo též *sítí*. *Ohodnocení hran* je zobrazení $a : E \to \mathbb{R}$, které přiřazuje hranám tyto doplňkové informace, což jsou např. doba trvání, délka úseku, pravděpodobnost událostí či cena za přepravu. Příklad ohodnoceného grafu je na obr. 2.2



Obrázek 2.2: Ohodnocený graf

Definice 2.7 Řekneme, že graf $G = (V, E)$ je *úplný*, jestliže mezi každými dvěma vrcholy $x, y \in V$, kde $x \neq y$, existuje alespoň jedna hrana e, která je spojuje. Ukázka úplného grafu je na obr. 2.3.



Obrázek 2.3: Úplný graf

2.2. Reprezentace grafů

Grafy lze často prezentovat graficky (viz obr. 2.1 - 2.3). Vrcholy obvykle znázorňujeme jako body nebo kroužky, hrany kreslíme jako čáry (většinou úsečky nebo oblouky). Je-li hrana orientovaná, používáme šipku od počátečního do koncového vrcholu. Grafy lze většinou zobrazit mnoha způsoby. Na první pohled nemusí být zřejmé, že se jedná o různá zakreslení téhož grafu.

Grafy však můžeme zobrazit také jinak než pomocí geometrických prostředků. Tuto možnost nejčastěji využíváme u větších grafů nebo v případě, že graf zadáváme do počítače.

Graf lze popsat podle definice dvěma množinami, tj. množinou vrcholů, která obsahuje výčet prvků, a množinou hran, jež je popsána seznamem dvojic vrcholů, případně seznamem trojic, pokud k dvojici vrcholů přidáme i jméno hrany. U orientovaných grafů záleží na pořadí vrcholů, jako první bývá uváděn počáteční vrchol, druhý údaj značí koncový vrchol. U neorientovaných grafů je pořadí vrcholů zvoleno libovolně. [9]

Běžnější ale bývá popis pomocí matic.

Definice 2.8 Nechť $G = (V, E)$ je graf s n vrcholy. *Matice sousednosti* grafu G je čtvercová $n \times n$ matice $A = (a_{ij})_{i,j=1}^{n}$ definovaná předpisem

$$a_{ij} = \begin{cases} 1 & \text{pro } \{v_i, v_j\} \in E, \\ 0 & \text{jinak.} \end{cases} \tag{2.1}$$

Příklad matice sousednosti z obr. 2.1 (a)

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Definice 2.9 Nechť $G = (V, E, \epsilon)$ je orientovaný graf bez smyček. Pokud pevně zvolíme pořadí n vrcholů a m hran, můžeme grafu G přiřadit *matici incidence* typu $n \times m$ předpisem

$$a_{ij} = \begin{cases} -1 & \text{jestliže } v_i \text{ je počátečním vrcholem hrany } e_j, \\ 1 & \text{jestliže } v_i \text{ je koncovým vrcholem hrany } e_j, \\ 0 & \text{v ostatních případech.} \end{cases} \tag{2.2}$$

Příklad matice incidence z obr. 2.1 (b)

$$A = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

3. Optimalizace

Postup, díky kterému dosáhneme „nejlepšího" řešení určitého problému, nazýváme optimalizace. Pokud je úloha formulována matematicky, označíme postup jako matematická optimalizace (optimem rozumíme maximum nebo minimum). [31]

Optimalizační úlohy se řeší od prvopočátku matematiky. Mnoho významných přírodovědců a matematiků bylo přesvědčeno, že chování přírody je optimální. Například Euler tvrdil: „*Na světě se nestane nic, v čem by nebylo vidět smysl nějakého maxima nebo minima.*" Leibnitz: „*Náš svět je nejlepší ze všech možných světů, a proto lze jeho zákony vyjádřit extremálními principy.*"[32]

K největšímu rozvoji optimalizace došlo ale až ve 20. století, především po druhé světové válce, kdy metody optimalizace pronikly do různých oblastí techniky, vědy a ekonomiky. [31]

Nyní si definujeme základní pojmy týkající se optimalizace. Čerpat budeme z [4], [12], [17], [23], [35] a [39].

Definice 3.1 Obecnou optimalizační úlohu lze formulovat ve tvaru:
minimalizuj

$$f(\mathbf{x}) \tag{3.1}$$

za podmínky

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0}, \tag{3.2}$$

kde $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ je vektor dimenze $n, x_j, j = 1, ..., n$ jsou tzv. rozhodovací proměnné, $f : \mathbb{R}^n \to \mathbb{R}$ je *účelová funkce*, jejíž extrém hledáme. Funkce $g_1 : \mathbb{R}^n \to \mathbb{R}, ...,$ $g_m : \mathbb{R}^n \to \mathbb{R}$ nazveme *omezujícími podmínkami.*

Potom množina $\mathbf{X} = \{\mathbf{x} | g_1(\mathbf{x}) \leq 0, ..., g_m(\mathbf{x}) \leq 0\}$ je označována jako *množina přípustných řešení.*

Transformací účelové funkce lze maximalizační úlohu převést na minimalizační:

$$\max_{\mathbf{x}} \{f(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\} = -\min_{\mathbf{x}} \{-f(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}. \tag{3.3}$$

O existenci extrémů pojednává následující Weierstrassova věta:

Věta 3.1 Nechť neprázdná množina přípustných řešení $\mathbf{X}$ je kompaktní (tj. ohraničená a uzavřená), pak spojitá účelová funkce $f(\mathbf{x})$ definovaná na této množině nabývá na ní globálního minima i maxima.

Definice 3.2 Řekneme, že funkce $f : \mathbb{R}^n \to \mathbb{R}$ má v bodě $\mathbf{x}_0 \in \mathbb{R}^n$ *lokální maximum* (resp. *lokální minimum*), jestliže existuje okolí $\mathbb{O}_{(\mathbf{x}_0)}$ takové, že $\mathbb{O}_{(\mathbf{x}_0)} \in Dom f$ a zároveň pro $\forall x \in \mathbb{O}_{(\mathbf{x}_0)}$ platí:

$$f(\mathbf{x}) \leq f(\mathbf{x}_0), \text{resp.} f(\mathbf{x}) \geq f(\mathbf{x}_0). \tag{3.4}$$

Analogicky je definováno *ostré lokální maximum* (resp. *ostré lokální minimum*), pro které platí:

$$f(\mathbf{x}) < f(\mathbf{x}_0), \text{resp.} f(\mathbf{x}) > f(\mathbf{x}_0). \tag{3.5}$$

Definice 3.3 Řekneme, že funkce $f : \mathbb{R}^n \to \mathbb{R}$ má v bodě $\mathbf{x}_0 \in \mathbb{R}^n$ *globální maximum* (resp. *globální minimum*), jestliže $\forall \mathbf{x} \in Dom f$ platí:

$$f(\mathbf{x}) \leq f(\mathbf{x}_0), \text{resp.} f(\mathbf{x}) \geq f(\mathbf{x}_0). \tag{3.6}$$

Pro *ostré globální maximum* (resp. *ostré globální minimum*), platí:

$$f(\mathbf{x}) < f(\mathbf{x}_0), \text{resp.} f(\mathbf{x}) > f(\mathbf{x}_0). \tag{3.7}$$

Optimalizační úlohy často dělíme na lineární a nelineární. V lineárních úlohách musí být omezení i účelová funkce lineární, nelineární úlohy zahrnují i jiné - nelineární vztahy.

Kvadratické programování

Pokud je účelová funkce polynomem druhého stupně a omezení jsou lineární, nazývá se řešený problém kvadratické programování. Čerpat budeme z [42]. Kvadratické programování budeme využívat v kapitole 6.

Definice 3.4 Úlohu kvadratického programování lze formulovat ve tvaru:

$$\min_{\mathbf{x} \in X} \{\frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x}\} \tag{3.8}$$

za podmínek

$$A\mathbf{x} \leq \mathbf{b}, \tag{3.9}$$

$$\mathbf{x} \in X \subset \mathbb{R}^n, \tag{3.10}$$

kde $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ je vektor dimenze $n, x_j, j = 1, ..., n$ jsou tzv. rozhodovací proměnné, $\mathbf{c}$ je konstantní váha lineární části, H je konstantní symetrická matice, která určuje váhu kvadratické části modelu. Chování modelu velmi závisí na charakteru matice H. Nejsnazší případ kvadratického programování vzniká v případě, že matice H je pozitivně definitní (pro problém s maximalizací negativně definitní). V tomto případě je účelová funkce konvexní (viz Definice 3.6) a optimální řešení je jedinečné.

Definice 3.5 Množina $\mathbf{X} \subseteq \mathbb{R}^n$ se nazývá *konvexní*, jestliže $\forall \mathbf{x}, \mathbf{y} \in \mathbf{X}, \forall \lambda \in [0, 1]$: $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in \mathbf{X}$. Množina $\{\lambda\mathbf{x} + (1-\lambda)\mathbf{y} | 0 \leq \lambda \leq 1\}$ tvoří úsečku spojující body $\mathbf{x}$ a $\mathbf{y}$. Definice tedy říká, že množina je konvexní, jestliže s každými dvěma body obsahuje i úsečku, která je spojuje.

Definice 3.6 Mějme reálnou funkci $f : \mathbf{X} \to \mathbb{R}$, kde $\mathbf{X} \subset \mathbb{R}^n$ je neprázdná konvexní množina. Řekneme, že f je *konvexní funkcí* na $\mathbf{X}$ právě tehdy, když pro každé dva body $\mathbf{x_1}, \mathbf{x_2}$ z množiny $\mathbf{X}$ a pro libovolné $\lambda, 0 \leq \lambda \leq 1$ platí:

$$f(\lambda\mathbf{x_1} + (1-\lambda)\mathbf{x_2}) \leq \lambda f(\mathbf{x_1}) + (1-\lambda) f(\mathbf{x_2}). \tag{3.11}$$

Funkce je tedy konvexní, pokud její graf leží pod libovolnou sečnou. Platí-li nerovnost jako ostrá, pak je funkce $f(x)$ na množině $\mathbf{X}$ *ryze konvexní*.

4. Statistika

V rámci této kapitoly se budeme věnovat testům normality, dále testům na odlehlá pozorování a zmíněna bude i shluková analýzy. Využívala se literatura [2], [45].

4.1. Testy normality

Některé statistické testy lze použít pouze pro určitý typ rozdělení, ze kterého získaná data pocházejí. Proto je velmi důležité ověřit, zda jsou dané předpoklady splněny. [6] Uvedeme si zde dva testy - Kolmogorovův-Smirnovův a Lillieforsův test. V případě prvního zmíněného testu budeme čerpat z [6], v případě druhého testu z [28].

4.1.1. Kolmogorovův-Smirnovův jednovýběrový test

Věta 4.1 Testujeme hypotézu, která tvrdí, že náhodný výběr $X_1, X_2, ..., X_n$ pochází z rozdělení s distribuční funkcí $\Phi(x)$. Nechť $F_n(x)$ je výběrová distribuční funkce. Testová statistika je tvaru:

$$D_n = \sup_{-\infty<x<\infty} |F_n(x) - \Phi(x)|. \tag{4.1}$$

Nulovou hypotézu zamítáme na hladině významnosti α, pokud $D_n \geq D_n(\alpha)$, kde $D_n(\alpha)$ je tabelovaná kritická hodnota. Pro $n \geq 30$ lze $D_n(\alpha)$ aproximovat výrazem 4.2.

$$D_n(\alpha) \doteq \sqrt{\frac{1}{2n} ln \frac{2}{\alpha}}. \tag{4.2}$$

Poznámka. Nulová hypotéza musí specifikovat distribuční funkci zcela přesně, včetně všech jejích případných parametrů.

4.1.2. Lillieforsova varianta

Pokud předem neznáme parametry normálního rozdělení, tj. střední hodnotu a rozptyl, můžeme použít Lillieforsovu variantu Kolmogorovova-Smirnovova testu. Místo přesných hodnot využíváme odhad střední hodnoty a rozptylu. Střední hodnotu odhadneme výběrovým průměrem

$$\tilde{\mu} = \overline{x} = \frac{1}{n} \sum_{i=1}^{n} X_i, \tag{4.3}$$

kde X jsou data a n je rozsah souboru. Rozptyl odhadneme výběrovým rozptylem

$$\tilde{\sigma}^2 = s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \tilde{\mu})^2. \tag{4.4}$$

Testová statistika je dána vztahem

$$D = \max_x |F^*(x) - S_N(x)|, \tag{4.5}$$

kde $S_N(x)$ je empirická distribuční funkce výběru a $F^*(x)$ je distribuční funkce $N(\tilde{\mu}, \tilde{\sigma}^2)$. Hypotézu o normalitě výběru zamítáme na hladině významnosti α, pokud $D \geq D(\alpha)$,

kde $D(\alpha)$ je tabelovaná kritická hodnota. Tabulku kritických hodnot lze nalézt v [28]. Hodnoty větší než 20 a menší než 30, které nejsou v tabulce uvedeny, byly dopočítány lineární interpolací, tedy pomocí vztahu

$$p(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0). \tag{4.6}$$

4.2. Testy na odlehlá pozorování

V souboru se mohou vyskytnout hodnoty, které se velmi liší od většiny ostatních a tím ovlivňují vlastnosti celého souboru. Tyto hodnoty označujeme jako odlehlá pozorování (anglicky outliers). [22] Vznikají z různých důvodů, vyjmenujeme si ty nejčastější. [25]

- **Náhoda.** Některá data se mohou od střední hodnoty lišit i o více než dvojnásobek směrodatné odchylky, přesto se nemusí jednat o chybné pozorování.
- **Chyba měření.** K chybě dojde kvůli technické vadě přístroje nebo kvůli selhání experimentátora.
- **Nevhodně zvolený soubor.** Výskyt určité hodnoty je velmi nepravděpodobný, ale přesto možný. Tato hodnota byla vybrána do základního souboru.
- **Chyba v zaznamenávání dat.** Chyba je způsobena přepsáním se při zápisu hodnot.

Velmi důležité je odlehlá pozorování odhalit. Někdy se v souboru vyskytují extremální hodnoty, které ale nepatří mezi odlehlá pozorování a pro další analýzu mohou nést důležité informace. Vždy je tedy nutné rozhodnout, zda lišící se data přijmout nebo je odmítnout. Zcela jistě chybnou hodnotu je třeba opravit nebo zahodit, aby nedošlo ke zkreslení celého souboru dat. Spolu s chybnou hodnotou musíme zahodit také všechna další pozorování, která z této hodnoty vycházela.

Pro určení odlehlých pozorování používáme různé testy a metody. Pro normální rozložení můžeme vypočítat aritmetický průměr a směrodatnou odchylku z dat bez podezřelé hodnoty. Jestliže vzdálenost podezřelé hodnoty od vypočítaného aritmetického průměru je větší než trojnásobek vypočítané směrodatné odchylky, jedná se pravděpodobně o odlehlé pozorování [27].

Dále si uvedeme dva testy (Grubbsův a Dean-Dixonův), které testují hypotézu, že hodnota není odlehlá, proti hypotéze, že se jedná o odlehlé pozorování.

Pokud testy odhalí odlehlé pozorování, provedeme je opakovaně se zbylým souborem dat.

4.2.1. Grubbsův test

V následujícím textu budeme čepat z [19].

Grubbsův test se používá pro testování odlehlých hodnot u normálního rozdělení. Rozsah souboru, tedy počet všech prvků souboru, musí být minimálně tři (v tom případě nesmí být žádné hodnoty stejné). Nejprve musíme sestavit hodnoty výběrového souboru do vzestupné řady, tj. $x_1 < x_2 < ... < x_n$.

Test na jedno odlehlé pozorování

Pro odhalení jednoho odlehlého pozorování používáme testovou statistiku T. Pokud se od ostatních hodnot liší nejmenší hodnota, použijeme:

$$T_1 = \frac{\overline{x} - x_1}{s}. \tag{4.7}$$

Pro případ, že předpokládané odlehlé pozorování je maximální hodnotou souboru, použijeme:

$$T_n = \frac{x_n - \overline{x}}{s}. \tag{4.8}$$

V případě, že nedokážeme rozhodnout, která z hodnot by mohla být odlehlým pozorováním, využijeme vztahu

$$T = max\{T_1, T_n\}. \tag{4.9}$$

$\overline{x}$ je aritmetický průměr (viz 4.3) a s je výběrová směrodatná odchylka (viz 4.10), která se vypočítá z výběrového rozptylu (viz 4.4) jako jeho odmocnina.

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}. \tag{4.10}$$

Vypočtené hodnoty T pak porovnáme s tabelovanou kritickou hodnotou T_α pro zvolenou hladinu významnosti α. Jestliže je vypočtená hodnota větší než T_α, jedná se o odlehlé pozorování. Tabulku kritických hodnot pro všechny typy Grubbsova testu lze nalézt v [18].

Grubbs ve své práci také odvodil, že hodnoty T_1 a T_n se dají vypočítat také pomocí následujících vztahů:

$$\frac{S_1^2}{S^2} = \frac{\sum_{i=2}^{n}(x_i - \overline{x}_1)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = 1 - \frac{T_1^2}{n-1}, \qquad \frac{S_n^2}{S^2} = \frac{\sum_{i=1}^{n-1}(x_i - \overline{x}_n)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} = 1 - \frac{T_n^2}{n-1}. \tag{4.11}$$

$\overline{x}_1, \overline{x}_n$ označují výběrové aritmetické průměry s vynecháním x_1, resp. x_n, tj.

$$\overline{x}_1 = \frac{1}{n-1} \cdot \sum_{i=2}^{n} x_i, \qquad \overline{x}_n = \frac{1}{n-1} \cdot \sum_{i=1}^{n-1} x_i. \tag{4.12}$$

V dalších testech se používají obměny těchto vztahů.

Test na jedno odlehlé pozorování na obou chvostech

V případě odlehlého pozorování na obou chvostech se testová statistika počítá podle vztahu:

$$\frac{S_{1,n}^2}{S^2} = \frac{\sum_{i=2}^{n-1}(x_i - \overline{x}_{1,n})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \tag{4.13}$$

kde $\overline{x}_{1,n}$ je bráno jako výběrový aritmetický průměr s vynecháním x_1 a x_n, tj.

$$\overline{x}_{1,n} = \frac{1}{n-2} \cdot \sum_{i=2}^{n-1} x_i. \tag{4.14}$$

Vypočtenou hodnotu pak porovnáme s tabelovanou kritickou hodnotou pro zvolenou hladinu významnosti α. Jestliže je vypočtená hodnota menší než tabelovaná, jedná se o odlehlé pozorování.

Test na více odlehlých pozorování na jednom chvostu

Pro dvě odlehlá pozorování využíváme podobné vztahy jako v testu na jedno odlehlé pozorování.

$$\frac{S_{1,2}^2}{S^2} = \frac{\sum_{i=3}^{n}(x_i - \overline{x}_{1,2})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \qquad \frac{S_{n-1,n}^2}{S^2} = \frac{\sum_{i=1}^{n-2}(x_i - \overline{x}_{n-1,n})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \tag{4.15}$$

kde $\overline{x}_{1,2}$ a $\overline{x}_{n-1,n}$ jsou brány jako výběrové aritmetické průměry s vynecháním x_1 a x_2, resp. x_{n-1} a x_n, tj.

$$\overline{x}_{1,2} = \frac{1}{n-2} \cdot \sum_{i=3}^{n} x_i, \qquad \overline{x}_{n-1,n} = \frac{1}{n-2} \cdot \sum_{i=1}^{n-2} x_i. \tag{4.16}$$

Stejně jako v předchozích případech vypočtenou hodnotu porovnáme s tabelovanou kritickou hodnotou pro zvolenou hladinu významnosti α. Jestliže je vypočtená hodnota menší než tabelovaná, jedná se o odlehlé pozorování.

Podobně pro k odlehlých pozorování počítáme podle vztahů

$$L'_k = \frac{\sum_{i=k+1}^{n}(x_i - \overline{x}'_k)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \qquad L_k = \frac{\sum_{i=1}^{n-k}(x_i - \overline{x}_{n-k})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}, \tag{4.17}$$

kde $\overline{x}_k$ a $\overline{x}_{n-k}$ jsou brány jako výběrové aritmetické průměry s vynecháním k prvních (nejnižších), resp. k posledních (nejvyšších) pozorování, tj.

$$\overline{x}'_k = \frac{1}{n-k} \cdot \sum_{i=k+1}^{n} x_i, \quad \overline{x}_{n-k} = \frac{1}{n-k} \cdot \sum_{i=1}^{n-k} x_i. \tag{4.18}$$

Vypočtenou hodnotu porovnáme s tabelovanou kritickou hodnotou pro zvolenou hladinu významnosti α. Jestliže je vypočtená hodnota menší než tabelovaná, jedná se o odlehlé pozorování.

4.2.2. Dean-Dixonův test

V následujícím textu budeme čepat z [10].

Tento test nepředpokládá normální rozdělení, test tedy můžeme použít, pokud neznáme rozdělení souboru. Stejně jako u Grubbsova testu musí být rozsah souboru minimálně tři (v tom případě nesmí být žádné hodnoty stejné). Opět je třeba sestavit hodnoty výběrového souboru do vzestupné řady, tj. $x_1 < x_2 < ... < x_n$.

Test na jedno odlehlé pozorování

Pokud se od ostatních hodnot liší nejmenší hodnota, použijeme:

$$r_{10} = \frac{x_2 - x_1}{x_n - x_1}. \tag{4.19}$$

Pro případ, že předpokládané odlehlé pozorování je maximální hodnotou souboru, použijeme:

$$r'_{10} = \frac{x_n - x_{n-1}}{x_n - x_1}. \tag{4.20}$$

V případě, že nedokážeme rozhodnout, která z hodnot by mohla být odlehlým pozorováním, vypočítáme r_{10} i r'_{10} a jako podezřelé odlehlé pozorování dále uvažujeme to, u kterého je výsledek větší.

Vypočtené hodnoty r_{10}, resp. r'_{10} pak porovnáme s tabelovanou kritickou hodnotou pro zvolenou hladinu pravděpodobnosti α. Tabulky kritických hodnot nalezneme v [11]. Pro soubory o velikosti větší než 30 a menší nebo rovno 100 byly kritické hodnoty vzaty z [43]. Jestliže je vypočtená hodnota větší než tabelovaná, jedná se o odlehlé pozorování.

Pokud chceme z testování vyloučit vliv některých hodnot, protože by mohly ovlivnit výsledek o odlehlosti, používáme následující vztahy:

$$r_{11} = \frac{x_2 - x_1}{x_{n-1} - x_1}, \qquad \text{resp.} \qquad r'_{11} = \frac{x_n - x_{n-1}}{x_n - x_2} \tag{4.21}$$

testuje odlehlost pozorování x_1 při vyloučení vlivu x_n, resp. odlehlost x_n při vyloučení x_1.

$$r_{12} = \frac{x_2 - x_1}{x_{n-2} - x_1}, \qquad \text{resp.} \qquad r'_{12} = \frac{x_n - x_{n-1}}{x_n - x_3} \tag{4.22}$$

testuje odlehlost pozorování x_1 při vyloučení vlivu x_n a x_{n-1}, resp. odlehlost x_n při vyloučení x_1 a x_2.

$$r_{20} = \frac{x_3 - x_1}{x_n - x_1}, \qquad \text{resp.} \qquad r'_{20} = \frac{x_n - x_{n-2}}{x_n - x_1} \tag{4.23}$$

testuje odlehlost pozorování x_1 při vyloučení vlivu x_2, resp. odlehlost x_n při vyloučení x_{n-1}.

$$r_{21} = \frac{x_3 - x_1}{x_{n-1} - x_1}, \qquad \text{resp.} \qquad r'_{21} = \frac{x_n - x_{n-2}}{x_n - x_2} \tag{4.24}$$

testuje odlehlost pozorování x_1 při vyloučení vlivu x_2 a x_n, resp. odlehlost x_n při vyloučení x_{n-1} a x_1.

$$r_{22} = \frac{x_3 - x_1}{x_{n-2} - x_1}, \qquad \text{resp.} \qquad r'_{22} = \frac{x_n - x_{n-2}}{x_n - x_3} \tag{4.25}$$

testuje odlehlost pozorování x_1 při vyloučení vlivu x_2, x_{n-1} a x_n, resp. odlehlost x_n při vyloučení x_{n-1}, x_1 a x_2.

Pro soubory o rozsahu 3-7 se používá většinou typ r_{10}, u souborů o rozsahu 8-10 typ r_{11}, pro rozsahy 11-13 typ r_{21} a u rozsahů větších než 14 použijeme r_{22} [25].

4.3. Shluková analýza

Shluková analýza je statistická metoda, díky které můžeme nalézt shluky (podmnožiny) určité množiny. Ve shluku jsou objekty s podobnými vlastnostmi, mimo shluk leží objekty, které se svými vlastnostmi odlišují.

Existuje několik různých metod, jak objekty shlukovat. Hlavní rozdělení je na hierarchické a nehierarchické. [34]

Pro naši případovou studii bude potřeba rozdělit data do různých shluků, abychom mohli provést statistické zpracování. V rámci této bakalářské práce byla zvolena nehierarchická metoda, konkrétně algoritmus k-means.

Více o shlukové analýze se lze dočíst například v [1], [14] a [37].

4.3.1. Algoritmus k-means

Na začátku zvolíme k shluků a k dat označíme jako tzv. centroidy. Algoritmus uvažuje data jako body v eukleidovském prostoru a v případě eukleidovské metriky, kterou budeme používat, se snaží minimalizovat vzdálenost mezi těmito body a centroidy. V každé iteraci tedy dojde ke změně složení shluků na základě minimální vzdálenosti bodů od centroidů podle zvolené metriky. Centroid se vždy po přidání dalšího bodu do shluku (příp. odebrání bodu) přepočítává. Algoritmus končí, když nedojde k přesunu žádného prvku.

Nevýhodou tohoto algoritmu je, že není zaručeno nalezení globálního minima. Výsledek totiž závisí na počáteční volbě centroidů.

5. Využívané programy a nástroje

5.1. Microsoft Excel, VBA

Microsoft Excel (MS Excel) je nejrozšířenějším tabulkovým kalkulátorem na světě. Můžeme ho používat nejen pro tvorbu tabulek, ale také přehledů, seznamů a databází a k organizaci dat. [5]

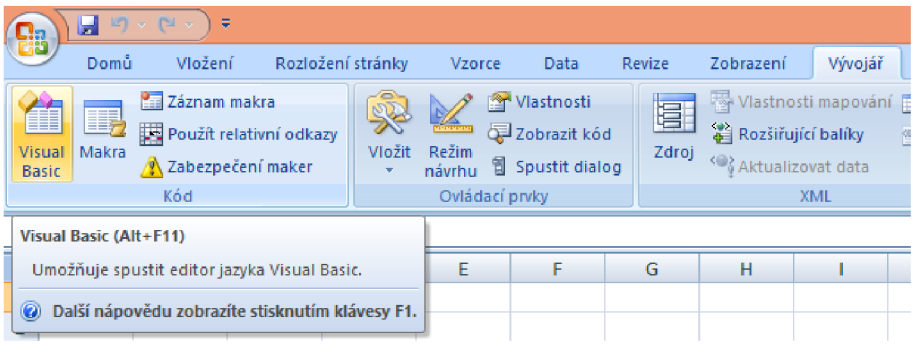
MS Excel je součástí balíku MS Office. Má uživatelsky příjemné prostředí, proto jej mohou využívat i méně zkušení uživatelé. Ve své práci jsem používala verzi MS Excel 2007.

MS Excel obsahuje možnost tvorby maker pomocí programovacího jazyku Visual Basic for Applications (VBA). V tomto prostředí byly vytvořeny skripty pro statistické testování dat (viz kap. 4.1, 4.2).

Visual Basic for Applications

Jediný programovací jazyk, který kancelářský balík MS Office podporuje, je VBA. Jedná se o objektově orientovaný jazyk, díky němuž je možné vytvářet makra. Makro je procedura, do které se zapisuje programový kód jako posloupnost příkazů. Používá se především ke zrychlení a zautomatizování rutinních prací. Makro lze vyvolat klepnutím na vytvořené tlačítko nebo ho můžeme přiřadit k určité klávesové zkratce. [3]

K editoru VBA se dostaneme přes kartu Vývojář, jak je znázorněno na obr. 5.1.



Obrázek 5.1: Umístění editoru Visual Basic

5.2. STATISTICA

STATISTICA je software, který obsahuje prostředky pro statistické zpracování dat. Je možné zde nalézt různé analýzy, testy, vizualizaci výsledků aj. [40] Součástí je i tzv. data mining, který odhaluje dříve neznámé vztahy mezi daty, a to pomocí procesu výběru, prohledávání a modelování ve velkých objemech dat. [24] Díky tomu byla v programu STATISTICA provedena shluková analýza.

5.3. GAMS

GAMS je program určený pro úlohy lineárního, nelineárního a míšeného celočíselného programování. Má vlastní programovací jazyk s poměrně uživatelsky přívětivou syntaxí.

GAMS má několik integrovaných řešičů a obsahuje kompilátor matematického jazyka. Můžeme díky němu vytvářet obsáhlé modely, jež se dokáží přizpůsobit novým situacím a podmínkám.

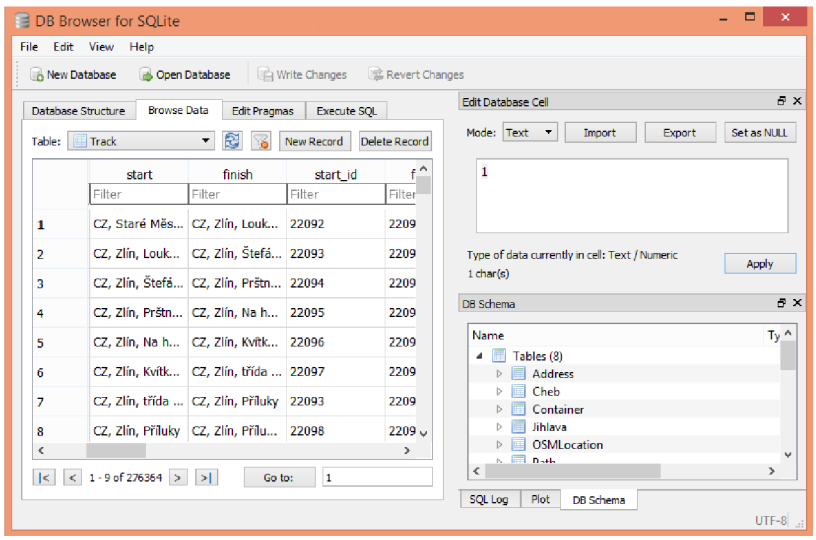
GAMS lze legálně stáhnout z internetových stránek www.gams.com. Informace o práci v tomto programu nalezneme v [36].

5.4. Databáze SQLite

SQLite je relaxační databázový systém obsažený v softwarové knihovně. Je šířen pod licencí public domain, lze jej tedy volně užívat. SQLite není založen na pricipu klient/server, takže neběží samostatně. Každá databáze s daty se ukládá do samostatného souboru na disku. S databázemi pracujeme prostřednictvím jazyka SQL [38].

5.4.1. DB Browser for SQLite

Pro vytvoření databáze byl využit nástroj DB Browser for SQLite. Díky němu můžeme vytvářet, upravovat a mazat tabulky a záznamy, které jsou součástí databáze. Uživatelské rozhraní je zobrazeno na obr. 5.2.



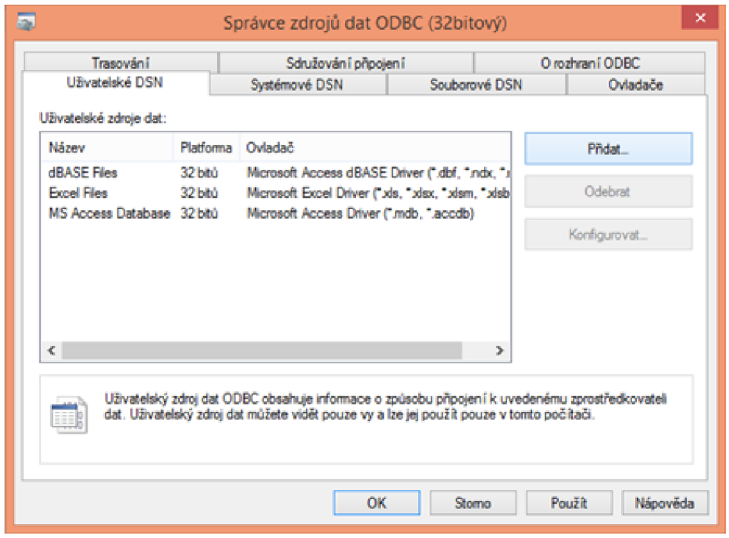
Obrázek 5.2: Uživatelské rozhraní nástroje DB Browser for SQLite

Do tabulek vytvořených v tomto nástroji byly zapsány adresy, úseky, trasy a další informace. Díky SQL dotazu pak mohla být vybrána ta data, která bylo třeba nahrát do programů Excel a STATISTICA, kde byla dále zpracovávána.

5.4.2. ODBC Driver

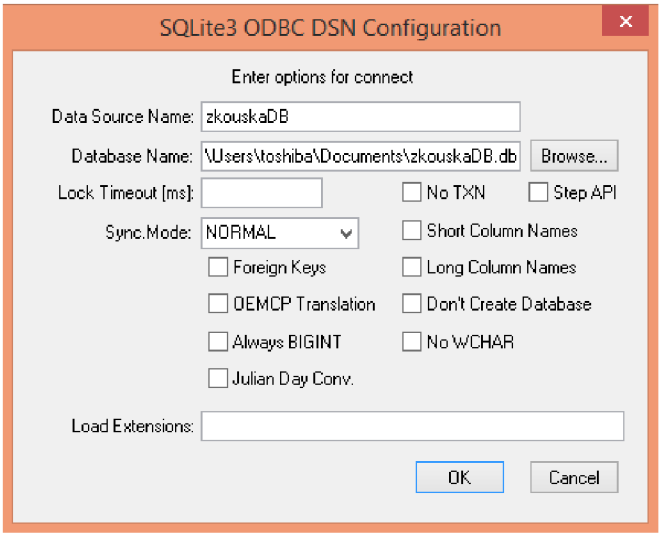
Pro propojení databáze SQLite a programu GAMS bylo třeba nainstalovat ODBC Driver. Tento nástroj slouží k propojení databáze a systému Windows (příp. Linux), díky čemuž můžeme načítat údaje z databáze do GAMSu. Z webových stránek http://www.ch-werner.de/sqliteodbc/ byl stažen soubor sqliteodbc.exe.

Po nainstalování se zobrazí uživatelské rozhraní (viz obr. 5.3).



Obrázek 5.3: Uživatelské rozhraní

Přidáme nový zdroj dat, a to konkrétně SQLite3 ODBC Driver. Následně si tento zdroj dat pojmenujeme a vybereme cestu k vytvořené databázi, kterou budeme chtít propojit se systémem GAMS (viz obr. 5.4).



Obrázek 5.4: Pojmenování zdroje dat

Tímto se vytvoří nový zdroj dat, který budeme dále využívat.
Podrobný popis propojení databází a systému GAMS lze najít v [7].

5.4.3. Propojení databáze SQLite a GAMSu

Pokud je potřeba nahrát do GAMSu řadu dotazů ze stejné databáze, bylo by volání SQL2GMS při každém dotazu neefektivní. Proto může být provedeno více dotazů a SQL2GMS stačí vyvolat pouze jednou. Kód, který se vloží před definováním množin, může vypadat například takto:

```
$ onecho > cmd.txt
C=DSN=zkouskaDB; dbq=zkouskaDB.db
```

```
Q1=SELECT distinct(Trasa) FROM GAMS GROUP BY Trasa
O1=trasy.inc

Q2=SELECT distinct(Usek) FROM GAMS GROUP BY Usek
O2=useky.inc

Q3=SELECT * FROM GAMS
O3=matice.inc

$ offecho
$ call=sql2gms @cmd.txt

Set
i trasy /
$include trasy.inc
/

j úseky /
$include useky.inc
/
;

Parameters
a(i,j) matice přiřazení úseku j k trase i /
$include matice.inc
/
;
```

Na druhém řádku kódu je napsaný vytvořený zdroj dat a název databáze i s příponou. Dále je v Q napsán SQL dotaz a O značí název include file.

6. Aplikační část

Mnoho firem a společností se musí zabývat dopravními a logistickými problémy. Řeší například, jakou zvolit trasu, aby vozidla projela určitá města nebo určité úseky. Z finančního hlediska je také potřeba určit optimální množství vozidel v návaznosti na kapacitě přepravovaného nákladu. Pro zásobovací firmy je také důležité umístění skladů.

Dopravní úlohy se tedy uplatňují například u poštovních služeb [41], u školní autobusové dopravy [44], při údržbě silnic (například úklid sněhu) [30] nebo při svozu odpadu [13].

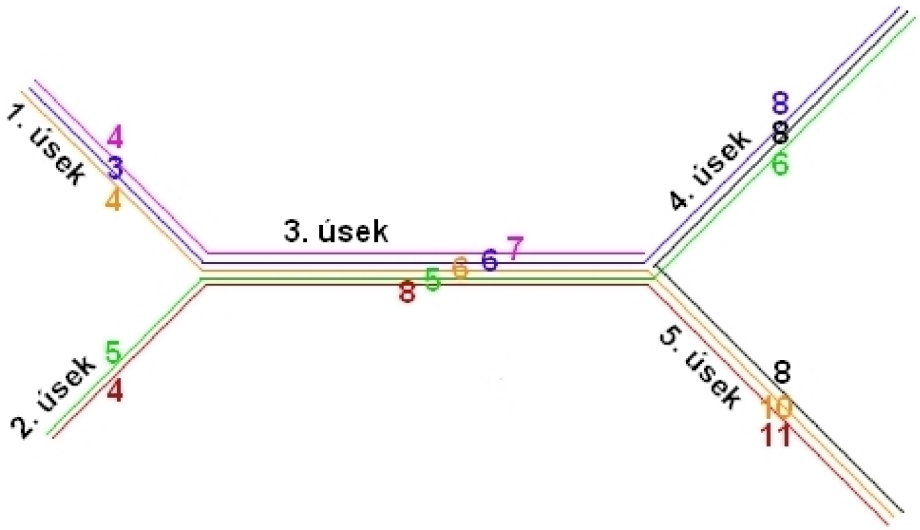
6.1. Ilustrativní příklad

Uveďme si modelovou situaci podle obr. 6.1. Barevně jsou vyznačeny jednotlivé trasy jízdy dopravních prostředků - trasa č. 1 má modrou barvu, druhá trasa je vyznačena oranžově, třetí trasa zeleně, čtvrtá hnědou barvou, pátá černou a šestá trasa fialově.

V příkladu je vyznačeno celkem pět úseků, které jsou na obrázku popsány.

Aby mohla být testována správná funkce matematického modelu, byl zvolen obrácený postup, než který využíváme v reálných datech. Konkrétně budeme předpokládat znalost doby průjezdu jednotlivých úseků pro konkrétní trasy. V reálném příkladu se tyto časy budeme snažit zjistit.

Na obr. 6.1 jsou uvedena čísla, která udávají dobu v minutách strávenou na jednotlivých úsecích. Barvy čísel odpovídají barvám tras, ke kterým se údaje o době váží. Průměrnou dobu strávenou na konkrétním úseku budeme značit t_j^*. Pro vyšší přehlednost jsou údaje, které vstupují do matematického modelu (viz kap. 6.2), vypsány v Tab. 6.1.



Obrázek 6.1: Schéma tras

Jak již bylo řečeno, reálná data často obsahují pouze časy průjezdu celých tras, nikoli jednotlivých úseků. Při znalosti existujících vazeb (součet časů průjezdu úseků je roven času průjezdu celé trasy) a dostupných dat (doba projetí trasy) je možné dílčí parametry jednotlivých úseků odhadnout. To je cílovým výstupem této reálné případové studie.

č. trasy	barva	úseky	celkový čas [min]
1	modrá	1, 3, 4	17
2	oranžová	1, 3, 5	20
3	zelená	2, 3, 4	16
4	hnědá	2, 3, 5	23
5	černá	4, 5	16
6	fialová	1, 3	11

Tabulka 6.1: Informace o trasách

K tomuto účelu byl vytvořen matematický model (kap. 6.2), který se snaží neznámé parametry (čas průjezdu úseků) odhadnout.

Do programu GAMS byl poté zadán celkový čas strávený na cestě a cílem programu bylo zjistit dobu strávenou na jednotlivých úsecích. Pro různé trasy mohou být doby průjezdu těchto úseků různé. Výsledná doba na úseku by se tedy měla pohybovat někde mezi časy uvedenými pro konkrétní úsek a jednotlivé trasy v ilustrativním příkladu.

6.2. Matematický model

Nejprve si definujme množiny (sets), parametry (parameters) a proměnné (variables), jež budeme potřebovat nejen v ilustrativním příkladě, ale také v hlavní případové studii (kap. 6.3).

Sets

$i \in I$	indexová množina tras
$j \in J$	indexová množina úseků

Parameters

T_i	celkový čas strávený na trase i
A_{ij}	matice přiřazení úseku j k trase i
t_{max_j}	maximální doba strávená na úseku j
t_{min_j}	minimální doba strávená na úseku j
$\bar{t}_j$	aritmetický průměr dob strávených na úseku j
w_j	váha - záleží na četnosti provozu
v_j	váha - záleží na variabilitě dat

Variables

δ_{ij}	odchylka času
z	účelová funkce
t_{ij}	doba strávená na úseku j trasy i
t_{0j}	optimální doba strávená na úseku j
p^+_{max}	kladná penalizace za překročení maximální doby
p^-_{max}	záporná penalizace za překročení maximální doby
p^+_{min}	kladná penalizace za nedosažení minimální doby
p^-_{min}	záporná penalizace za nedosažení minimální doby

Nejprve byl sestaven idealizovaný model, díky kterému bylo možné odhalit, jaké další omezující podmínky budou třeba přidat.

Matematický model měl následující tvar:

$$\text{minimalizuj} \quad z = \sum_{i,j} \delta_{ij}^2 \tag{6.1}$$

za podmínek

$$T_i = \sum_j (A_{ij} \cdot t_{ij}), \forall i \in I, \tag{6.2}$$

$$t_{0j} = t_{ij} + \delta_{ij}, \forall i \in I, j \in J, \tag{6.3}$$

$$t_{ij} \geq 0, \forall i \in I, j \in J, \tag{6.4}$$

$$t_{0j} \geq 0, \forall j \in J. \tag{6.5}$$

Cílem účelové funkce(rovnice 6.1) je zajistit vyvážený odhad časů průjezdů t_{0j} pomocí minimalizace kvadrátu odchylek δ_{ij}. První omezující podmínka (6.20) dává do souvislosti celkový čas na trase i a časy pro jednotlivé úseky j této trasy. Celkový čas se tedy rovná součtu časů všech úseků, kterými daná trasa vede. Z důvodu, že přes jeden úsek j vede více tras i, není zaručeno, že čas strávený na konkrétním úseku bude pro všechny trasy stejný, tj. $t_{kj} = t_{lj}$ pro $k \neq l; k, l \in I$. Hledaná doba průjezdu úseku t_{0j} se rovná součtu doby průjezdu tohoto úseku t_{ij} a odchylky δ_{ij}. To vyjadřuje rovnice 6.21. Odchylka času je tedy rozdílem hledané doby průjezdu úseku (tato doba je stejná pro všechny trasy) a doby průjezdu úsekem pro konkrétní trasu. Rovnice 6.4 říká, že doba strávená na úseku j trasy i musí být nezáporná. Stejně tak je nezáporná i hledaná doba průjezdu úseku j, což vyjadřuje poslední omezující podmínka (6.5).

Ověření modelu na zjednodušeném příkladu

Vstupní data budeme čerpat z ilustrativního příkladu z kap. 6.1. Výsledky úlohy nedokázaly na základě matematického modelu odhalit časové nároky na jednotlivé úseky. To je dáno tím, že chceme získat minimální celkovou chybu danou prvky δ_{ij}. Program GAMS proto přiřadil některým úsekům poměrně malý čas, konkrétně úsekům jedna a dva. Bylo tedy potřeba model upravit. Výsledky prvního idealizovaného modelu jsou zobrazeny v příloze této bakalářské práce, pro přehlednost byly přepsány také do Tab. 6.2.

	Trasa	**Úsek**				
		1	**2**	**3**	**4**	**5**
chyba	1	-0,271	0	-0,271	-0,271	0
	2	0,396	0	0,396	0	0,396
	3	0	0,333	0,333	0,333	0
	4	0	-0,333	-0,333	0	-0,333
	5	0	0	0	-0,063	-0,063
	6	-0,125	0	-0,125	0	0
$\mathbf{t_{0j}}$	-	0,396	1,208	10,354	5,437	10,438
$\mathbf{t_j^*}$	-	3,667	4,5	6,4	7,333	9,667

Tabulka 6.2: Výsledky idealizovaného modelu [min]

Byla přidána omezení, která zajistí reálnost výsledků. Jednalo se o maximální, resp. minimální dobu strávenou na úseku (viz Tab. 6.3). Tato doba byla stanovena s ohledem na vstupní data ilustrativního příkladu (viz obr. 6.1). Při překročení, resp. nedosažení, této doby došlo k penalizaci.

č. úseku	t_{min}	t_{max}
1	3	6
2	4	5
3	5	9
4	6	9
5	8	12

Tabulka 6.3: Maximální a minimální doba [min]

Přidané podmínky měly následující tvar:

$$p^{+}_{max;ij} - p^{-}_{max;ij} = t_{max;j} - t_{ij}, \forall i \in I, j \in J, \tag{6.6}$$

$$p^{+}_{min;ij} - p^{-}_{min;ij} = t_{ij} - t_{min;j}, \forall i \in I, j \in J, \tag{6.7}$$

$$p^{+}_{min;ij} \geq 0, \forall i \in I, j \in J, \tag{6.8}$$

$$p^{-}_{min;ij} \geq 0, \forall i \in I, j \in J, \tag{6.9}$$

$$p^{+}_{max;ij} \geq 0, \forall i \in I, j \in J, \tag{6.10}$$

$$p^{-}_{max;ij} \geq 0, \forall i \in I, j \in J. \tag{6.11}$$

Rovnice 6.22 říká, že pokud překročíme maximální dobu strávenou na úseku j (tato doba je uvedena v Tab. 6.3), dojde k penalizaci. Podobně rovnice 6.23 říká, že pokud nedosáhneme minimální doby strávené na úseku j (tato doba je uvedena v Tab. 6.3), opět dojde k penalizaci. Rovnice 6.8 - 6.11 vyjadřují, že penalizace jsou nezáporné.

Tvar účelové funkce 6.1 se změnil na tvar 6.19.

$$z = \sum_{j} (\frac{w_j^2}{v_j} \cdot \sum_{i} {\delta_{ij}}^2) + \sum_{i,j} ({p^{-}_{max;ij}}^2 + {p^{-}_{min;ij}}^2), \tag{6.12}$$

kde w je váha, která vyjadřuje, že záleží na počtu tras, které daným úsekem vedou. Bez váhy w_j (resp. $w_j = 1$) by pro frekventovanější úseky, kde je více odchylek δ_{ij}, model více dbal na volbu vyvážené hodnoty t_{0j}. Pokud bychom chtěli, aby na četnosti provozu záleželo ještě více, tzn. aby vyvážený odhad t_{0j} byl blízký průměrné hodnotě t_{ij} pro všechny trasy i, je vhodné počítat váhu w_j podle vztahu 6.13.

$$w_j = \frac{\sum_{i} A_{ij}}{\max_{j}(\sum_{i} A_{ij})}, \forall j \in J \tag{6.13}$$

Pokud bychom naopak žádali, aby chyby δ_{ij} nezávisely na tom, jak je úsek frekventovaný, mohli bychom váhu určit podle vztahu 6.14.

$$w_j = \frac{1}{\sum_i A_{ij}}, \forall j \in J \tag{6.14}$$

V našem případě jsme chtěli, aby na četnosti provozu záleželo, ale nechtěli jsme frekventovaným úsekům dávat příliš velkou váhu, proto bylo zvoleno $w_j = 1$.

Je vhodné, aby chyba závisela také na variabilitě dat na konkrétním úseku. V rámci výpočtu bude potřeba určit aritmetický průměr časů pro různé trasy a daný úsek:

$$\bar{t}_j = \frac{\sum_{i=1}^{n} t_{ij}}{|I|}, \forall j \in J, \tag{6.15}$$

kde $|I|$ značí kardinalitu množiny I (tj. počet prvků množiny I).

Úloha se řeší iteračně. Nejprve se odhadne t_{ij} bez váhy v_j (tedy $v_j = 1$) a následně se přepočítají váhy v_j. Váhu v_j dopočítáme jako rozptyl.

$$v_j = \frac{\sum_{i=1}^{n} (\bar{t}_j - t_{ij})^2}{|I|}, \forall j \in J. \tag{6.16}$$

Výsledky pak odpovídaly ilustrativnímu příkladu (viz 6.1), jak je patrné z Tab. 6.4. Tyto výsledky jsou pak také zobrazeny v příloze.

	Trasa	**Úsek**				
		1	**2**	**3**	**4**	**5**
chyba	1	-0,094	0	-0,389	-0,046	0
	2	0,139	0	0,577	-0,003	0,339
	3	0	0,436	0,574	0,059	0
	4	0	-0,436	-0,574	-0,003	-0,337
	5	0	0	0	-0,004	-0,002
	6	-0,045	0	-0,189	-0,003	0
$\mathbf{t_{0j}}$	-	3,966	4,564	6,800	5,706	10,289
$\mathbf{t_j^*}$	-	3,667	4,5	6,4	7,333	9,667

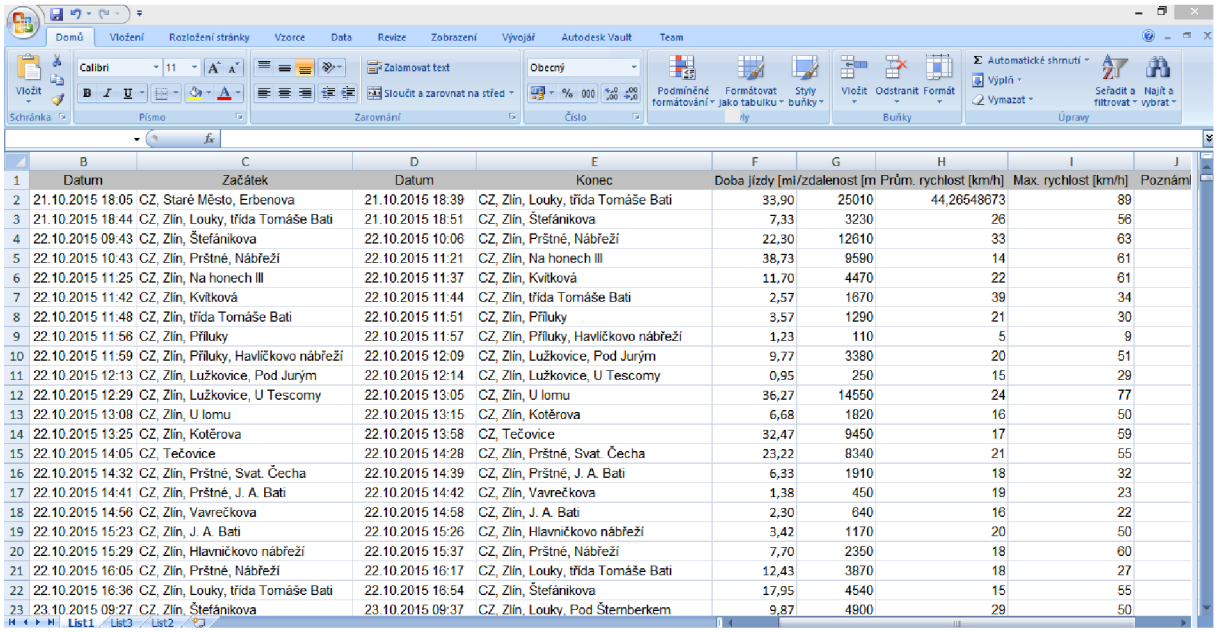
Tabulka 6.4: Výsledky modifikovaného modelu [min]

6.3. Případová studie

Jak již bylo řečeno v kap. 6.1, cílem případové studie je zjistit dobu průjezdu jednotlivých úseků tras. Celkově můžeme řešení daného problému rozdělit do následujících osmi bodů.

1. Příprava dat

Pro statistické zpracovaní mi byl dodán soubor, který obsahoval 325 193 řádků. Ukázka souboru je na obr. 6.2. V každém řádku byla uvedena místa, odkud vozidlo vyjelo a kam jelo, dále čas odjezdu a příjezdu, vzdálenost, kterou vozidlo urazilo, maximální a průměrná rychlost během jízdy a identifikace vozidla. Data byla získána od firem, které disponují zařízeními, jež tyto informace zaznamenávají.



	B	C	D	E	F	G	H	I	J
1	Datum	Začátek	Datum	Konec	Doba jízdy [mi	zdalenost [m	Prům. rychlost [km/h]	Max. rychlost [km/h]	Poznámi
2	21.10.2015 18:05	CZ, Staré Město, Erbenova	21.10.2015 18:39	CZ, Zlín, Louky, třída Tomáše Bati	33,90	25010	44,26548673	89	
3	21.10.2015 18:44	CZ, Zlín, Louky, třída Tomáše Bati	21.10.2015 18:51	CZ, Zlín, Štefánikova	7,33	3230	26	56	
4	22.10.2015 09:43	CZ, Zlín, Štefánikova	22.10.2015 10:06	CZ, Zlín, Prštné, Nábřeží	22,30	12610	33	63	
5	22.10.2015 10:43	CZ, Zlín, Prštné, Nábřeží	22.10.2015 11:21	CZ, Zlín, Na honech III	38,73	9590	14	61	
6	22.10.2015 11:25	CZ, Zlín, Na honech III	22.10.2015 11:37	CZ, Zlín, Kvítková	11,70	4470	22	61	
7	22.10.2015 11:42	CZ, Zlín, Kvítková	22.10.2015 11:44	CZ, Zlín, třída Tomáše Bati	2,57	1670	39	34	
8	22.10.2015 11:48	CZ, Zlín, třída Tomáše Bati	22.10.2015 11:51	CZ, Zlín, Příluky	3,57	1290	21	30	
9	22.10.2015 11:56	CZ, Zlín, Příluky	22.10.2015 11:57	CZ, Zlín, Příluky, Havlíčkovo nábřeží	1,23	110	5	9	
10	22.10.2015 11:59	CZ, Zlín, Příluky, Havlíčkovo nábřeží	22.10.2015 12:09	CZ, Zlín, Lužkovice, Pod Jurým	9,77	3380	20	51	
11	22.10.2015 12:13	CZ, Zlín, Lužkovice, Pod Jurým	22.10.2015 12:14	CZ, Zlín, Lužkovice, U Tescomy	0,95	250	15	29	
12	22.10.2015 12:29	CZ, Zlín, Lužkovice, U Tescomy	22.10.2015 13:05	CZ, Zlín, U lomu	36,27	14550	24	77	
13	22.10.2015 13:08	CZ, Zlín, U lomu	22.10.2015 13:15	CZ, Zlín, Kotěrova	6,68	1820	16	50	
14	22.10.2015 13:25	CZ, Zlín, Kotěrova	22.10.2015 13:58	CZ, Tečovice	32,47	9450	17	59	
15	22.10.2015 14:05	CZ, Tečovice	22.10.2015 14:28	CZ, Zlín, Prštné, Svat. Čecha	23,22	8340	21	55	
16	22.10.2015 14:32	CZ, Zlín, Prštné, Svat. Čecha	22.10.2015 14:39	CZ, Zlín, Prštné, J. A. Bati	6,33	1910	18	32	
17	22.10.2015 14:41	CZ, Zlín, Prštné, J. A. Bati	22.10.2015 14:42	CZ, Zlín, Vavrečkova	1,38	450	19	23	
18	22.10.2015 14:56	CZ, Zlín, Vavrečkova	22.10.2015 14:58	CZ, Zlín, J. A. Bati	2,30	640	16	22	
19	22.10.2015 15:23	CZ, Zlín, J. A. Bati	22.10.2015 15:26	CZ, Zlín, Hlavničkovo nábřeží	3,42	1170	20	50	
20	22.10.2015 15:29	CZ, Zlín, Hlavničkovo nábřeží	22.10.2015 15:37	CZ, Zlín, Prštné, Nábřeží	7,70	2350	18	60	
21	22.10.2015 16:05	CZ, Zlín, Prštné, Nábřeží	22.10.2015 16:17	CZ, Zlín, Louky, třída Tomáše Bati	12,43	3870	18	27	
22	22.10.2015 16:36	CZ, Zlín, Louky, třída Tomáše Bati	22.10.2015 16:54	CZ, Zlín, Štefánikova	17,95	4540	15	55	
23	23.10.2015 09:27	CZ, Zlín, Štefánikova	23.10.2015 09:37	CZ, Zlín, Louky, Pod Šternberkem	9,87	4900	29	50	

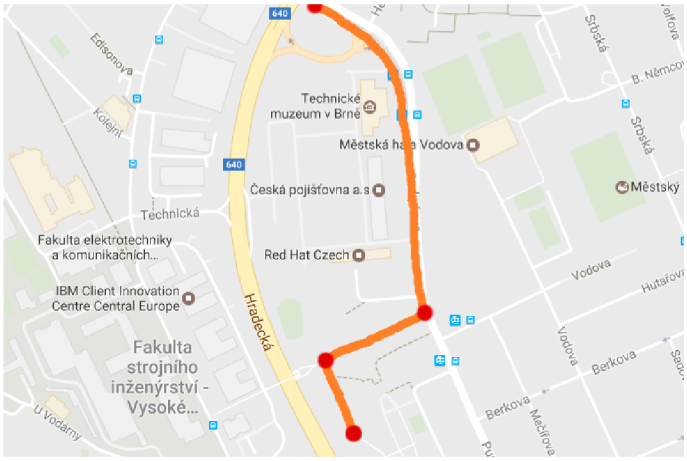
Obrázek 6.2: Ukázka souboru s daty

Data však bylo potřeba upravit - smazat nelogické hodnoty a hodnoty, které nelze dále zpracovávat, jako je například záporný či nulový čas na cestě, záporná rychlost či shodné místo odjezdu a příjezdu. Tak došlo ke snížení počtu použitelných dat na 276 966 řádků.

2. Hledání trasy

Další práce byla prováděna v databázovém systému SQLite. Protože v souboru z MS Excel byla informace pouze o místě odjezdu a příjezdu, byla pravděpodobná trasa určena na základě mapového algoritmu - OpenStreetMap (OSM) podle nejkratší cesty. Tyto mapy jsou k dispozici zdarma, přesto je pokrytí měst velmi dobré. Pomocí OSM byly zvoleny počáteční a koncové body tras.

Dále bylo třeba rozdělit tyto trasy na úseky. Opět pomocí OSM byly zvoleny určité uzly, jako jsou křižovatky, které jednotlivé úseky oddělují. Příklad úseků a uzlů je na obr. 6.3.



Obrázek 6.3: Vyznačení úseků (oranžová barva) a uzlů (červené body)

Celé hledání trasy bylo prováděno na základě práce [29], kde jsou popsány všechny algoritmy a kde lze nalézt informace o OSM, popis výběru uzlů aj.

Jakmile byly známy úseky, ze kterých se trasa skládá, mohla být v databázi SQLite vytvořena tabulka, která obsahovala identifikační čísla a propojení tras a úseků. Po nahrání této tabulky do GAMSu byla vytvořena matice sousednosti, jež přiřazuje trase úseky, ze kterých se skládá.

3. Charakter dat

Následně bylo třeba k úsekům přiřadit informace, o jaký typ cesty se jedná. Typem cesty se myslí například dálnice, silnice první, druhé a třetí třídy. Tento krok probíhal rovněž pomocí OSM.

Každý úsek k sobě měl tedy přiřazenu jednu z následujících informací (čerpáno z [8]):

- motorway	dálnice
- trunk	rychlostní komunikace
- prim	silnice I. třídy
- secondary	silnice II. třídy
- tertiary	silnice III. třídy
- road	dočasné označení silnice neznámé třídy
- residential	místní komunikace v obci
- service	účelová komunikace
- motorway-link	nájezdy a sjezdy k dálnicím
- trunk-link	nájezdy a sjezdy k rychlostním komunikacím
- primary-link	nájezdy a sjezdy k silnicím I. třídy
- secondary-link	nájezdy a sjezdy k silnicím II. třídy
- tertiary-link	nájezdy a sjezdy k silnicím III. třídy
- living-street	komunikace v obytné zóně
- unclassified	jiné nezařazené

4. Výběr použitelných dat

Všechny tyto údaje byly zapsány do tabulek v databázi. Tam se shromažďovaly všechny potřebné informace. Následně pomocí SQL dotazů mohly být vybrány pouze ty, které byly důležité k dalšímu zpracování. Tato data byla importována zpět do Excelu.

V dalším kroku bylo třeba vyfiltrovat trasy, u kterých vzdálenost, jež mělo vozidlo urazit podle map, odpovídala skutečné vzdálenosti. K dalšímu zpracování se tedy vzaly jen ty trasy, u kterých byl poměr skutečné vzdálenosti a vzdálenosti podle map od 0,9 do 1,1. Celkem tedy zbylo 84 610 tras. Obrovské snížení použitelných dat může být dáno tím, že podle map byla hledána nejkratší cesta (viz bod 2). Vozidlo však mohlo zvolit jinou trasu, například z důvodu nehody. Dále mohlo hrát roli například mýto atd. Dále mohly vzniknout problémy například ztrátou signálu GPS během jízdy, proto bylo zaznamenáno jiné místo, než kde se vozidlo skutečně nacházelo. Ke zvýšení použitelných dat by například pomohlo, kdybychom neznali pouze místo odjezdu a příjezdu, ale i průjezdné body trasy.

Aby nedocházelo k tak velkým ztrátám použitelných dat, vyžaduje tato část práce další rozvoj a je cílem dalších činností v rámci vývoje výpočtového systému.

5. Shluková analýza

Vstupní data bylo třeba dále očistit o odlehlé hodnoty. Pro použití následujících testů (bod 6 a 7) bylo nutné shlukovat trasy s podobným charakterem (viz bod 3). V programu STATISTICA byla provedena shluková analýza pomocí algoritmu k-means. Počet shluků byl zvolen tisíc. Startovací hodnoty výpočtu (centroidy) byly zvoleny náhodně. Výpočet vzdáleností probíhal na bázi Euklidovské metriky. Shlukování bylo prováděno podle procentuálního zastoupení různých typů cest (z kolika procent se trasa skládá z dálnic, silnic první, druhé, třetí třídy, atd.; viz bod 3) a podle délky trasy.

Vzniklo tedy tisíc shluků, nejmenší z nich obsahoval sedm prvků, největší jich měl 276.

6. Test normality

Další výpočty probíhaly v Excelu pomocí vytvořeného makra ve VBA. Pro každý shluk byla tvořena distribuční funkce na základě průměrných rychlostí tras. Dále pro každý shluk mohla být odhadnuta střední hodnota a směrodatná odchylka pro rychlost. Díky tomu bylo možné otestovat hypotézu, zda data z daného shluku pochází z normálního rozdělení či nikoli. K tomuto účelu byl použil Lillieforsův test (viz 4.1.2). Celkem vyšlo 651 shluků z normálního rozdělení.

7. Grubbsův x Dean-Dixonův test

Na shluky s normálním rozdělením byl aplikován Grubbsův test na odlehlá pozorování. Nejprve byl proveden test na jedno odlehlé pozorování na jednom chvostu. Tento test byl opakován tak dlouho, dokud identifikoval extrém. Díky němu bylo zjištěno celkem 59 tras, u kterých se průměrná rychlost lišila od většiny ostatních v daném shluku natolik, že byla označena jako odlehlá. Tyto trasy by příliš ovlivňovaly další výsledky, proto byly odstraněny. Poté byl proveden test na dvě odlehlá pozorování na jednom chvostu, jenž měl zjistit, zda v souboru nejsou dvě odlehlé hodnoty, které nemohly být testem na jedno odlehlé pozorování odhaleny. I tento test byl prováděn opakovaně, dokud byly nalezeny odlehlé hodnoty. Díky tomu bylo odstraněno dalších 68 tras.

Podobně pro data, která nepochází z normálního rozdělení, byl proveden Dean-Dixonův test. Do makra ve VBA byly implementovány testy r_{10}, r_{11}, r_{21} a r_{22} a podle počtu prvků ve shluku byl zvolen vhodný test. Nejčastěji se jednalo o test r_{22}, protože počet prvků byl většinou větší než 14. Celkem bylo zjištěno 48 odlehlých pozorování. Stejně jako v případě Grubbsova testu byla tato data smazána.

8. Import dat do GAMSu

Všechny trasy, které prošly testy na odlehlá pozorování (bylo jich celkem 84 431), byly nahrány do databáze SQLite, kde k nim byly opět přiřazeny úseky (resp. jejich čísla), ze kterých se trasy skládají. Celkově se trasy skládaly z 172 322 úseků.

Data bylo třeba importovat do vytvořeného projektu v GAMSu. Propojení databáze a programu GAMS je popsáno v kap. 5.4.

Maximální, resp. minimální doba, strávená na úseku byla stanovena výpočtem, a to podle vztahu 6.17, resp. 6.18.

$$t_{max} = \frac{\text{délka úseku}}{\text{minimální rychlost pro daný typ úseku}} \tag{6.17}$$

$$t_{min} = \frac{\text{délka úseku}}{\text{maximální rychlost pro daný typ úseku}} \tag{6.18}$$

Maximální rychlost pro daný typ úseku byla stanovena jako maximální povolená rychlost. Minimální rychlost byla určena z praktických zkušeností (dle ÚPI). Tabulka s rychlostmi pro dané typy úseků je zobrazena v příloze.

6.4. Výsledky a možnosti dalšího vývoje

Aby mohl být program dále využíván, je třeba zhodnotit časovou náročnost výpočtu. Z důvodu příliš rozsáhlých dat (již dříve zmiňovaných 84 431 tras a 172 322 úseků) se řešení ukázalo výpočtově příliš náročné (program GAMS nahlásil chybu *Out of memory*). Pro snadnější iniciaci úlohy byly zvoleny startovací hodnoty pomocí lineárního modelu, který je významně snadněji řešitelný. Tvar jeho účelové funkce byl:

$$z = \sum_j \frac{w_j}{v_j} \cdot \sum_i ({\delta_{ij}}^+ + {\delta_{ij}}^-) + \sum_{i,j} (p^-_{max;ij} + p^-_{min;ij}) \tag{6.19}$$

a omezující podmínky:

$$T_i = \sum_j (A_{ij} \cdot t_{ij}), \forall i \in I, \tag{6.20}$$

$$t_{0j} = t_{ij} + {\delta_{ij}}^+ - {\delta_{ij}}^-, \forall i \in I, j \in J, \tag{6.21}$$

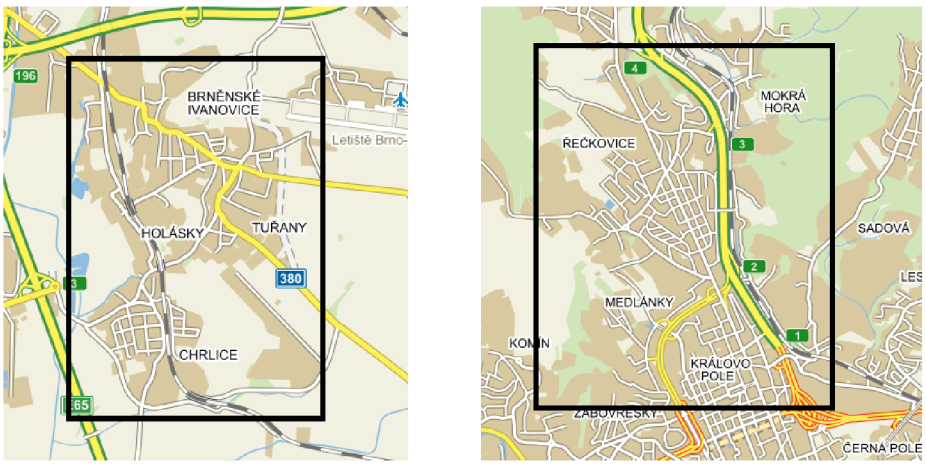
$$p^+_{max;ij} - p^-_{max;ij} = t_{max;j} - t_{ij}, \forall i \in I, j \in J, \tag{6.22}$$

$$p^+_{min;ij} - p^-_{min;ij} = t_{ij} - t_{min;j}, \forall i \in I, j \in J. \tag{6.23}$$

Proměnné $t_{ij}, {\delta_{ij}}^+, {\delta_{ij}}^-, t_{0j}, p^+_{min;ij}, p^-_{min;ij}, p^+_{max;ij}, p^-_{max;ij}$ jsou nezáporné pro $\forall i \in I, j \in J$. Pro ověření výpočtového systému byly zvoleny menší oblasti (viz obr. 6.4), pro které byl výpočet realizován.

Počet tras	112	1386
Počet úseků	2954	2847
Počet nenulových prvků t_{ij} v matici přiřazení	9110	50793
Časová náročnost načítání z databáze	45 s	1 min 2 s
Časová náročnost výpočtu	2 min 35 s	69 h 26 min 40 s

Výpočty probíhaly na počítači s těmito parametry:
Procesor: Intel(R)Xeon(R) CPU E5-2698 v4 @ 2.20GHz 2.20 GHz
Nainstalovaná paměť: 128 GB

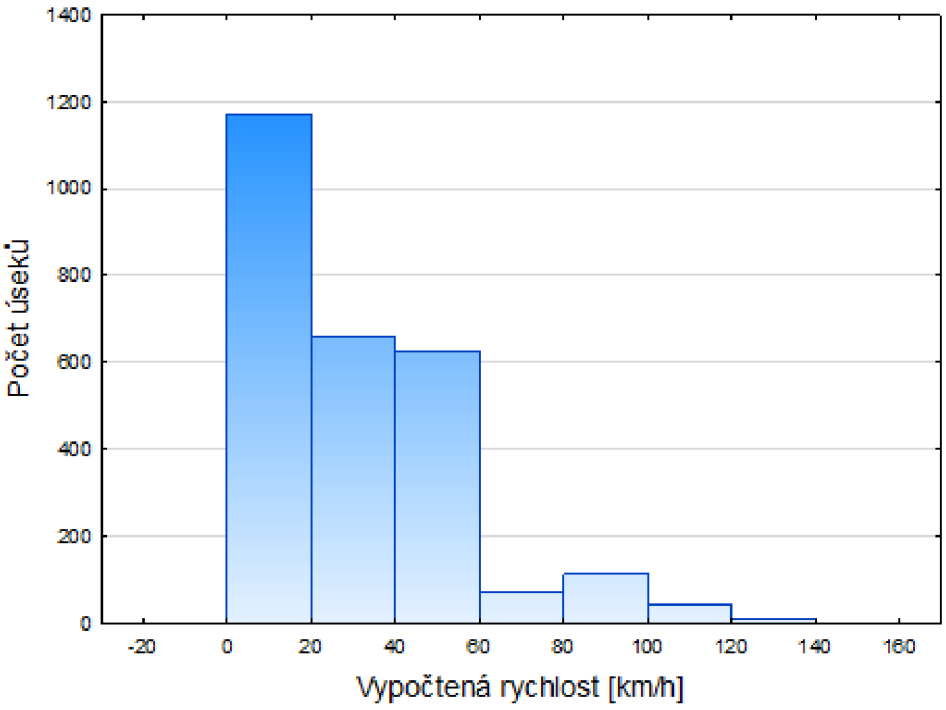


(a) (b)

Obrázek 6.4: (a) oblast se 112 použitelnými trasami, (b) oblast s 1386 použitelnými trasami

Typ systému: 64bitový operační systém pro platformu x64

Výsledky pro variantu b) byly zpracovány do histogramu, viz obr. 6.5. Zvolená oblast představovala městskou dopravní síť včetně dálnice a několika rychlostních silnic. Na obr. 6.5 jsou uvedeny odhadované rychlosti v km/h pro zvolenou oblast 6.4 b).



Obrázek 6.5: Histogram

Úloha pro celou Českou republiku je v přijatelném výpočtovém čase neřešitelná. Bylo by vhodné provést clustering, díky němuž by úloha byla rozdělena do několika menších oblastí a pro ně by byl proveden výpočet. Zmíněný postup je však již nad rámec této bakalářské práce, proto nebyl proveden.

Závěr

Tato práce byla zaměřena na zpracování velkého množství dat z reálného provozu.

V úvodní části byly uvedeny základní pojmy z teorie grafů, optimalizace a statistické testy. Dále byly v práci popsány programy a nástroje, které byly dále využívány. V této části bylo důležité propojení databáze se systémem GAMS, které bylo dále využíváno.

Stěžejní kapitolou celé práce je kapitola 6, která se věnuje aplikační práci. Po představení matematického modelu byla funkčnost ověřena na ilustrativním příkladu. Následně byla popsána hlavní případová studie. Vstupní data pro analyzovaný problém pocházejí z GPS zařízení různých dopravních společností. Tyto informace se liší v detailu zaznamenávání, tj. v prodlevě mezi jednotlivými záznamy v průběhu přepravy. Odhad časové náročnosti průjezdů úseků v požadovaném detailu, který je významně vyšší oproti záznamům z GPS, byl rozdělen do osmi kroků.

První byl pojmenován příprava dat. Z důvodu zjištění (odhadnutí) doby strávené na jednotlivých úsecích trasy, což je důležité pro řešení dopravních a logistických problémů a bylo to zároveň cílem této práce, bylo nutné obdržená data upravit a vyfiltrovat.

Dalším bodem bylo hledání trasy. Pomocí OSM byla zvolena trasa, kterou se vozidlo dostalo z počátečního do koncového místa. Tato trasa byla následně rozdělena do jednotlivých úseků, ke kterým byly přiřazeny další informace.

Charakter dat byl popsán ve třetím kroku. Čtvrtý bod se věnuje výběru použitelných dat, tato fáze probíhala v MS Excel. V dalším kroku bylo potřeba provést shlukovou analýzu (pátý krok) a následně byly prováděny statistické testy. V šestém bodě byl proveden test normality, díky kterému mohl být v sedmém kroku vybrán vhodný statistický test na odlehlé hodnoty (Grubbsův x Dean-Dixonův test).

V posledním, osmém kroku, byla data importována do GAMSu, který určil hledanou dobu průjezdu jednotlivých úseků. Pro rozsáhlé soubory je v aktuální podobě výpočet nemožný z důvodu nedostatku paměti. Pro malé oblasti dává výpočtový systém dobré výsledky za akceptovatelný čas výpočtu.

Data byla uchovávána v databázovém systému SQLite, proto by v případě potřeby bylo opětovné statistické zpracování snadné.

Literatura

[1] AGGARWAL, Charu C a Chandan K. REDDY. *Data clustering: algorithms and applications.* Boca Raton, FL: CRC Press, 2014. ISBN 978-1-4665-5821-2.

[2] ANDĚL, Jiří. *Statistické metody.* Vydání 2. Praha: Matfyzpress, 1998.

[3] ANDRÁSSY, Roland. *Visual Basic for Applications: V prostredí MS Office* [online]. 2003 [cit. 2017-04-14]. Strojnícka fakulta TU Košice. Dostupné z: http://www.sjf.tuke.sk/transferinovacii/pages/archiv/transfer/6-2003/pdf/148-150.pdf

[4] BENÁČKOVÁ, Jana. *Modelování energetického zdroje a plánování jeho provozu s využitím pokročilých matematických metod.* Brno: Vysoké učení technické v Brně, 2011. Vedoucí Ing. Martin Pavlas, Ph.D.

[5] BROŽA, Petr. *Microsoft Office 2007: [průvodce pro každého].* Brno: Zoner Press, 2007. ISBN 978-80-86815-58-9.

[6] BUDÍKOVÁ, Marie, Tomáš LERCH a Štěpán MIKOLÁŠ. *Základní statistické metody.* Brno: Masarykova univerzita v Brně, 2005. ISBN 80-210-3886-1.

[7] *Converting database tables to GAMS data.* [online]. [cit. 2017-04-28]. Dostupné z: http://lyle.smu.edu/emis/docs/GAMS/GDX/sql2gms.pdf

[8] *Cs: Map Features - OpenStreetMap Wiki.* [online]. [cit. 2017-04-28]. http://wiki.openstreetmap.org/wiki/Cs:Map_Features

[9] DEMEL, Jiří. *Teorie grafů.* Vyd. 2. přeprac. Praha: České vysoké učení technické, 1991. ISBN 80-01-00567-4.

[10] DIXON, Juan W. Analysis of extreme values. *The Annals of Mathematical Statistics.* **21**(4), 1950, 488-506.

[11] DIXON, Juan W. Ratios involving extreme values. *The Annals of Mathematical Statistics.* **22**(1), 1951, 68-78.

[12] DOSTÁL, Zdeněk a Petr BEREMLIJSKI. *Metody optimalizace.* [online]. 2012 [cit. 2017-01-28]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/metody_optimalizace.pdf

[13] EISELT, H. A., Michel GENDREAU a Gilbert LAPORTE. Arc Routing Problems, Part II: The Rural Postman Problem. *Operations Research.* [online]. **43**(2), 1995, 399-414. [cit. 2017-02-19]. Dostupné z: http://dx.doi.org/10.1287/opre.43.3.399

[14] EVERITT, Brian. *Cluster analysis.* Vydání 5. Chichester: Wiley, 2011. ISBN 978-0-470-74991-3.

[15] FERDAN, T., ŠOMPLÁK, R., ZAVÍRALOVÁ, L., PAVLAS, M., FRÝBA, L. A Waste-to-Energy Project: A Complex Approach towards the Assessment of Investment Risks. *Applied Thermal Engineering.* **89**(1), 2015, 1127-1136. ISSN: 1359-4311.

[16] FOLTÝNEK, Tomáš a Jana DANNHOFEROVÁ. *Teorie grafů*. Brno: Mendelova univerzita v Brně, 2011. ISBN 978-80-7375-500-3.

[17] GÁL, Tomáš. *Lineární programování*. Praha: Státní pedagogické nakladatelství, 1969.

[18] GRUBBS, Frank E. a Glenn BECK. Extension of Sample Sizes and Percentage Points for Significance Tests of Outlying Observations . *Technometrics*. **14**(4), 1972, 847-854.

[19] GRUBBS, Frank E. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics*. **21**(1), 1950, 27-58.

[20] HLINĚNÝ, Petr. *Základy teorie grafů*. [online].[cit. 2017-01-26]. Dostupné z: http://is.muni.cz/do/1499/el/estud/fi/js10/grafy/Grafy-text10.pdf

[21] CHRISTOFIDES, Nicos. *Graph theory: An Algorithmic Approach*. Third printing. London: Academic Press, 1975. ISBN 0-12-174350-0.

[22] JAROŠOVÁ, Eva a Darja NOSKIEVIČOVÁ. *Pokročilejší metody statistické regulace procesu*. Vydání 1. Praha: Grada Publishing, 2015. ISBN 978-80-247-5355-3.

[23] KLAPKA, Jindřich, Jiří DVOŘÁK a Pavel POPELA. *Metody operačního výzkumu*. Brno: Vysoké učení technické, 1996. ISBN 80-214-0817-0.

[24] KLÍMEK, Petr. Shlukovací metody v data miningu. *E+M Ekonomie a Management*. **11**(2), 2008, 120-127. ISSN 1212-3609.

[25] KOTLORZ, Lukáš. *Testy normality*. Praha: Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2012. Vedoucí prof. RNDr. Jiří Anděl, DrSc.

[26] KUL'KA, Jozef, Martin MANTIČ, Melichar KOPAS, Eva FALTINOVÁ a Daniel KACHMAN. Heuristic Optimization Approach to Selecting a Transport Connection in City Public Transport. *Open Engineering*. **7**(1), 1-5.

[27] LEHMANN, Rüdiger. 3 σ - Rule for Outlier Detection from the Viewpoint of Geodetic Adjustment. *Journal of Surveying Engineering*. **139**(4), 2013, 157–165.

[28] LILLIEFORS, Hubert W. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*. **62**(318), 1967, 399-402.

[29] LIPOVSKÝ, Tomáš. *Modelování rizik v dopravě*. Brno: Vysoké učení technické v Brně, Ústav soudního inženýrství, 2016. 64 s. Vedoucí diplomové práce RNDr. Pavel Popela, Ph.D.

[30] MENESES, Susana a Adelino FERREIRA. New Optimization Model for Road Network Maintenance Management. *Procedia - Social and Behavioral Sciences*. **54**(4), 2012, 956-965.

[31] MÍKA, Stanislav. *Matematická optimalizace*. Plzeň: ZČU Plzeň, 1997. ISBN 80-7082-319-4.

[32] MÍKA, Stanislav a Ludvík VLČEK. *Matematické optimalizační technologie*[online]. 2011 [cit. 2017-01-28]. Dostupné z: http://home.zcu.cz/~mika/MO/S_Mika_L_Vlcek_MOT.pdf

[33] NEŠETŘIL, Jaroslav a Jiří MATOUŠEK. *Kapitoly z diskrétní matematiky*. Praha: MATFYZPRESS, 1996. ISBN 80-85863-17-0.

[34] ONDERLIČKA, Tomáš. *Aplikace shlukové analýzy na reálných datech*. Brno: Vysoké učení technické v Brně, 2016. Vedoucí RNDr. Libor Žák, Ph.D.

[35] PYTELA, Oldřich. *Optimalizace*. Vydání 1. Pardubice: Vysoká škola chemickotechnologická, 1982.

[36] ROSENTHAL, Richard E. *GAMS: A User's Guide* [online]. 2017 [cit. 2017-04-14]. GAMS Development Corporation, Washington, DC, USA. Dostupné z: https://www.gams.com/latest/docs/userguides/GAMSUsersGuide.pdf

[37] ŘEZANKOVÁ, Hana, Dušan HÚSEK a Václav SNÁŠEL. *Shluková analýza dat*. Vydání 2., rozš. Praha: Professional Publishing, 2009. ISBN 978-80-86946-81-8.

[38] SQLite Tutorial. In: *SQLite Overview* [online]. [cit. 2017-04-14]. Dostupné z: http://www.tutorialspoint.com/sqlite/sqlite_tutorial.pdf

[39] SMEJKALOVÁ, Veronika. *Modely pro rekonstrukci toku v síti*. Brno: Vysoké učení technické v Brně, 2016. Vedoucí Ing. Martin Pavlas, Ph.D.

[40] *StatSoft CR s.r.o.: Ovládání a základy statistiky v softwaru STATISTICA*. [online]. [cit. 2017-04-28]. Dostupné z: http://www.statsoft.cz/file1/PDF/StrucnyManualSTATISTICA.pdf

[41] SUN, Li, ZHAO, Lindu a Jing HOU. Optimization of postal express line network under mixed driving pattern of trucks. *Transportation Research Part E: Logistics and Transportation Review*. **77**, 2015, 147–169.

[42] ŠANDERA, Čeněk. *Heuristic Algorithms in Optimization*. Brno: Vysoké učení technické v Brně, 2008. Vedoucí Ing. Jan Roupec, Ph.D.

[43] VERMA, Surendra P. a Alfredo QUIROZ-RUIZ. Critical values for six Dixon tests for outliers in normal samples up to sizes 100, and applications in science and engineering. *Revista Mexicana de Ciencias Geológicas*. **23**(2), 2006, 133–161.

[44] ZHANG, Jian-jun a Yue-guang LI. School Bus Problem and its Algorithm. *IERI Procedia*. **2**, 2012, 8-11.

[45] ZVÁRA, Karel a Josef ŠTĚPÁN. *Pravděpodobnost a matematická statistika*. Vydání 3. Praha: Matfyzpress, 2002. ISBN 80-85863-93-6.

A. Seznam použitých zkratek

GAMS	General Algebraic Modeling System
GPS	Global Positioning System
MS	Microsoft
ODBC	Open Database Connectivity
OSM	OpenStreetMap
SQL	Structured Query Language
ÚPI	Ústav procesního inženýrství
VBA	Visual Basic for Application

B. Přílohy

B.1. Výsledky modelů v GAMSu

```
----     60 VARIABLE delta.L  chyba

            1           2           3           4           5

1      -0.271 -1.9178E-16      -0.271      -0.271 -3.3213E-17
2       0.396 -3.5009E-17       0.396 7.79366E-16       0.396
3 -5.5511E-17       0.333       0.333       0.333 1.13163E-16
4 -5.5511E-17      -0.333      -0.333 1.01359E-15      -0.333
5 -5.5511E-17 1.24090E-19 -7.3184E-19      -0.063      -0.063
6      -0.125 4.80612E-18      -0.125 8.72097E-16 -8.0050E-17


----     60 VARIABLE t0.L  optimální doba strávená na úseku j

1  0.396,    2  1.208,    3 10.354,    4  5.437,    5 10.438


----     60 VARIABLE t.L  doba strávená na úseku j trasy i

            1           2           3           4           5

1       0.667       1.208      10.625       5.708      10.438
2                   1.208       9.958       5.437      10.042
3       0.396       0.875      10.021       5.104      10.438
4       0.396       1.542      10.687       5.437      10.771
5       0.396       1.208      10.354       5.500      10.500
6       0.521       1.208      10.479       5.437      10.438
```

Obrázek B.1: Výsledky idealizovaného modelu

```
----     84 VARIABLE delta.L  chyba

             1           2           3           4           5

1       -0.094 -1.0110E-12       -0.389       -0.046 -1.7524E-10
2        0.139 4.87121E-13        0.577       -0.003        0.339
3 -1.0360E-11        0.436        0.574        0.059 6.75757E-11
4 -6.0316E-12       -0.436       -0.574       -0.003       -0.337
5 1.63893E-11 1.85812E-13 3.52964E-16       -0.004       -0.002
6       -0.045 3.37178E-13       -0.189       -0.003 1.03610E-10


----     84 VARIABLE t0.L  optimální doba strávená na úseku j

1  3.966,    2  4.564,    3  6.800,    4  5.706,    5 10.289


----     84 VARIABLE t.L  doba strávená na úseku j na trase i

             1           2           3           4           5

1        4.060       4.564       7.189       5.752      10.289
2        3.827       4.564       6.223       5.709       9.950
3        3.966       4.128       6.226       5.646      10.289
4        3.966       5.000       7.374       5.709      10.626
5        3.966       4.564       6.800       5.709      10.291
6        4.011       4.564       6.989       5.709      10.289
```

Obrázek B.2: Výsledky modifikovaného modelu

B.2. Tabulka rychlostí pro typy dopravních úseků

typ úseku	v_{min}	v_{max}
Motorway	60	130
Trunk	60	130
Prim	50	110
Secondary	50	90
Tertiary	50	90
Road	20	90
Residential	20	50
Service	10	50
Motorway-link	50	80
Trunk-link	50	80
Primary-link	20	70
Secondary-link	20	70
Tertiary-link	20	70
Living-street	10	30
Unclassified	20	90

Tabulka B.1: Minimální a maximální rychlost pro různé typy úseků